DOC1: Saca casa
DOC2: Aca hay asas
DOC3: Casa asa saca
DOC4: Aca aca asta

a.

Recorro los documentos para obtener los términos a indexar, luego divido cada termino según el documento que corresponda:

Saca (DOC 1)
Casa (DOC 1)
Aca (DOC 2)
Hay (DOC 2)
Asas (DOC 2)
Casa (DOC 3)
Asa (DOC 3)
Saca (DOC 3)
Aca (DOC 4)
Aca (DOC 4)
Asta (DOC 4)

Ahora agrupo cada término según los documentos en los que se encuentra y diferenciando entre paréntesis la distancia a la que se encuentra el término desde la posición 1 (Primer término del documento). En naranja coloco la diferencia de las posiciones entre los términos que aparecen en varios documentos.

Saca -> Doc1 (1) (1), Doc3 (3)(2)
Casa -> Doc1 (2), Doc3 (1)
Aca -> Doc2 (1)(1), Doc4 (1,1)(3)
Hay -> Doc2 (2)
Asas -> Doc2 (3)
Asa -> Doc3 (2)
Asta -> Doc4 (3)

Ordeno alfabéticamente para poder hacer una búsqueda binaria luego, encerrando entre paréntesis el número del byte que corresponde para donde inicia el término :

Lo anterior no está almacenado en memoria.

Lexico : (0) aca (3) asa (6) asas (10) asta (14) casa (18) hay (21) saca

Codificando las distancias a documentos, la frecuencia con la que aparece el término en un documento y la posición que está en el doc en gamma:

*Punteros a doc: (0) 0101101001011 (13) 0111010 (20) 0101011 (27) 001001011 (36) 11010010111 (46) 0101010 (53) 1110101011

*Referencia para legibilidad en colores: distancia a documento, segundo la frecuencia del término en el doc, tercero la posición en el doc:

Marque en verde la posición del bit de inicio para cada palabra, esto es para legibilidad mia solamente

El índice invertido resulta:

Término(no va almacenado en memoria el término de esta columna en si, si el léxico.)	Puntero Léxico	Puntero Docs			
(0) Aca	0	0			
(1) Asa	3	13			
(2) Asas	6	20			
(3) Asta	10	27			
(4) Casa	14	36			
(5) Hay	18	46			
(6) Saca	21	53			

b) Q = “Saca casa”

Comienzo buscando el término “Saca”, mediante búsqueda binaria:

1. (Acceso a pos 3 del índice, **Acceso** a léxico, Acceso al siguiente índice para ver hasta donde leer, pos 4, voy del 10 a 13 inclusive.
   Leo “Asta”, no es el buscado, voy a la parte de abajo, Acceso a pos 5 del índice, Acceso al siguiente para ver nuevamente hasta donde leer, pos 6, voy del 18 al 20. **Acceso** al léxico pos 18 , “hay” no es el buscado, voy hacia abajo, Acceso a pos 6 del índice, **Acceso** al léxico, “saca” es el buscado, Acceso a puntero a docs bit 53 (estando en EOF), leo un 1 (doc1) aparece 1 vez en la pos 1. Luego leo 010 (a

distancia 2 del doc 1, por lo tanto es el doc3), aparece 1 vez en la pos 3.

-> Doc 1 (pos 1) y Doc 3 (pos 3)

2. Buscamos el término casa, Acceso(cacheada) a pos 3 del índice, ya accedida, **Acceso(cacheada)** al léxico, leo(ya cacheado, no lo cuento) hasta la pos del léxico siguiente al indice 3, de 10 a 13, no es el término buscado, voy hacia la parte inferior, Acceso(cacheada) a pos 5 del índice, al léxico, ya accedida, tampoco es el término buscado, voy a la parte superior, Acceso a pos 4 del índice, **Acceso** al léxico pos 14, leyendo hasta 17, es el buscado, Acceso a puntero bit 36, se mira dónde es que empieza el puntero a doc del próximo término, como ya accedimos (cacheado, al igual que los anteriores) leeremos los punteros a documentos del bit 36 a 45 inclusive.

   Leo un 1 doc 1, aparece 1 vez, posición 2. Avanzamos, leyendo 010, que es a distancia 2 del 1, osea doc 3, aparece 1 vez en la posición 1.

   -> Doc 1 (pos 2) y Doc 3 (pos 1)

Como la frase buscada está en ambos documentos, analizamos las posiciones en las que se encuentran para ver si se respeta la proximidad.

"casa" -> Doc 1 (pos 2) y Doc 3 (pos 1)
"saca" -> Doc 1 (pos 1) y Doc 3 (pos 3)

Como la query era "saca casa"
"saca" está en la pos 3 en el doc 3, pero "casa" está en la pos 1, entonces no respeta el orden.
Mientras que "saca" del doc 1 está en la pos 1, y "casa" en la pos 2, si respeta la proximidad. Entonces el **término consulta está** en el **documento 1**.

En total fueron 14 accesos, contando los cacheados.

c) Construyendo los bigramas de cada término del documento:

Saca = $s sa ac ca a$
Casa = $c ca as sa a$
Aca = $a ac ca a$
Hay = $h ha ay y$
Asas = $a as sa as s$
Asa = $a as sa a$
Asta = $a as st ta a$

$s -> Saca
sa -> Saca, Casa, Asas, Asa
ac -> Saca, Aca
ca -> Saca, Aca
a$ -> Saca, Casa, Aca,Asa,Asta
$c -> Casa
as -> Casa, Asas, Asa
$h -> Hay
ha -> Hay
ay -> Hay
y$ -> Ay
$a -> Aca, Asas, Asa, Asta
st -> Asta
ta -> Asta

Ordenando:

$a (0)
$c (2)
$h (4)
$s (6)
a$ (8)
ac (10)
as (12)
ay (14)
ca (16)
ha (18)
sa (20)
st (22)
ta (24)
y

Manteniendo el léxico anterior, construyo un índice secundario donde cada bigrama apunta a los términos que lo contengan, esto es construido cuando se lee cada término (mencionado al inicio del documento este)

	Distancia ptr a léxico	
0 ($s)	21	
1 (ac)	3,3,8,7	
2 (ac)	0,21	
3 (ca)	0,21	
4 (a$)	0,3,7,4,7	

5 ($c)		
6 (as)		
7 ($h)		

Administré mal el tiempo y no me dio para seguir completando la tabla, ni ordenar bien la lista puntero a término. Para resolver la consulta habría que verificar los bigramas, luego chequear en que documento se encuentra el termino que lo contiene en la tabla del ej a.

Q = as*a -> $a AND as AND a$
Osea, que empiecen con a, tenga el término as y luego que termine con a, una vez obtenido el documento que cumpla con esto habría que verificar que efectivamente se cumpla lo buscado porque puede haber un falso positivo.